UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/041,743	01/10/2002	Scott James Weaver	9288 (3225-130)	3989

26884 7590 03/04/2010
PAUL W. MARTIN
NCR CORPORATION, LAW DEPT.
1700 S. PATTERSON BLVD.
DAYTON, OH 45479-0001

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
03/04/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/041,743
Filing Date: January 10, 2002
Appellant(s): WEAVER, SCOTT JAMES

Stevens Harden III
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 12/15/2009 appealing from the Office action mailed 4/07/2009.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct for the most part, except for the part that matches the claimed subject matter being rejected under the USC 112 Rejection, whose deficiency will be presented in further details in the following Examiner's Answer.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

20030149934 Worden 8-2003

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

/**********************/

Claim Rejections - 35 USC § 112

1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. Claims 17-31 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Specifically, claim 17 and 25 recites 'creating a third model … in a data wedge by integrating the first schema and the second schema into a data wedge'; and according to using a data wedge instantiating a model data, represented as a tree visible to a respective user based on the user's schema individually submitted by the respective user, it is not recognized from the Specifications that an act of integrating 2 schemas takes place to yield a "third data model" as recited. The tree view instance (see Specifications: Fig. 1, para 0023,pg 6) to correspond to each user schema (data view A, view B) clearly describes that W3C content of schema is converted into a separate view visible to the respective user (see Specifications: own data model – para

0027, pg. 8; separate views – para –30, pg. 8; para 0033, pg. 9) according to his/her own provision of a model (schema A, schema B – Fig. 1). Thus, schema model A and schema model B are not deemed integrated into a view that have integrated content (emphasis added) of the 2 models. The *third model* being a result of an integration - integrating schema A **and** schema B - is deemed not in possession of the Inventor when the invention was made. One cannot make use of the so-recited integration step (to form a third data model), when there is no clear teaching enabling for this step so to obviate one of ordinary skill to apply undue experimentation. The Wedge is a physical environment or common tool (also described as a COM server having a integration purpose) whereas subscribed client's "data model" (or schema) submitted thereto are returned by the Wedge server as DOM type instance or a "tree" or "data view" as disclosed (middle pg. 7; bottom pg. 11; Drawings: Fig. 1-2) and this tree view amounts to a view visible to the client whose schema has been inputted via NW communication (see Fig. 1 and related text) and this visibility would enable each such client to add or modify that particular client-specific view (pg. 8, middle; Populate 34, Fig. 2). The term 'integrating' is deemed devoid of substance because the Specifications teach no particular algorithmic process by which elements of a schema are systematically integrated into the wedge so that the end result (e.g. tangible data model-like output based on inputs coming from separate schema elements) of this process reasonably conveys combined schema elements of at least two schema. That is, the Wedge data/tree view (analogized to this "third data model") is NOT one with integrating/merging result from respective schema elements of at least two users. Since no part in the Disclosure deliberately and explicitly describes that a DOM view (or third model distinct from first and second model) contains substantially or **all** XML elements coming from at least two (emphasis added) user's provided schemas. The limitation as to creating a third model from integrating 2 schemas (i.e. "creating a third data model … by integrating"), as identified above, will be given no real patentable weight but will be treated as though the plurality of schemas are inputs (from users subscribed to a server) into the integration server with multi-format mappings like a wedging/mapping framework and converted in some organized data hierarchy (or a graphical model-like constructs) visible to the respective user (schema submitting clients).

3. Likewise, claims 17, and 25 fail to comply with the written description requirement. Specifically, claims 17 and 25 recite: 'creating a first schema … ; creating a second schema …'. According to the Specifications, the data wedge receives schemas from users (see schema A, B - Fig. 1) and there is not a description therein that explicitly mentions about (the method/system being claimed as) a software capability to create the schema A or B as mentioned above. The very act of creating schema --prior to this schema is converted into a third model as claimed -- is not supported by a single description in the entire disclosure so to convey that the inventor actually possesses an established utility or programmatic means that would explicitly perform this creation as claimed. The inventor is not deemed in possession of a capability (with respect to the context of claims 17, 25) to create schemas prior to submitting these schemas into the data wedge environment, simply because schemas coming from (or generated by) users cannot be construed as work being done by the invention. The *creating* a first schema and second schema as identified above will be treated as receiving schema representing model associated with that schema.

Claims 18-24, 26-31 do not remedy to the above lack of support from the Disclosure, hence will be rejected for lack of proper description.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 17, 19, 21-25, 28-32 are rejected under 35 U.S.C. 102(e) as being anticipated by Worden, USPuBN: 2003/0149934 (hereinafter Worden).

As per claim 17, Worden discloses a computer-implemented method of storing and translating data between a format of a first data model of a first software component and a format of a second data model of a second software component, the method comprising:

creating a first schema comprising the first data model of the first software component; creating a second schema comprising the second data model of the second software component (see par 0030, pg. 2; para 0087-0098, pg. 4 - Schema (1) Schema (2) – Fig. 9 - Note: receiving schemas – see para 0035, 0040, 0042, pg. 3 -being created by the sending clients involved in a transaction – refer to USC 112, 1st para Rejection – see para 0090-0097, pg. 6 – where each such XML structure represents a model type business components – see Fig. 2, Fig. 58-68 – reads on first or second data model – see UML - para 0037, pg. 3; para 0098, pg. 6);

creating a third data model (para 0298-0301, pg. 16; Fig. 58-68) and a data storage in a data wedge by integrating the first schema and the second schema into the data wedge (see Fig. 9; para 0068-0070, pg. 5 –**Note1**: using XMulator --para 0239-0240, pg. 14 - to map business model components to XML specification from more than one clients – para 0038 pg. 3; Fig. 58-68 – the XML representing any intended languages or business applications, the mapping to derive model classes – see para 0030, pg. 2 - **reads on** first and second model schemas integrated into the *wedge* – see para 0029-0035, pg. 2-3 – **Note2**: Interoperability framework using UML methodology as basis for structuring/storing/representing data/objects defined for matching more than one user's XML schema submitted as data representation of the very user business model – para 0007-0010 - in terms of business class or model blocks – see *business information model* in Java classes - para 0324-0330, pg. 18; para 0239, pg. 13 - **reads on** third model created from integrating schemas into the XMulator environment and storage of thereof – Fig. 58-68 – as a result of mappings from the XMulator; that is, 'third data model' treated as mere a model like representation based on all schema submitted to the integrating or mapping service - see USC 112 Rejection);

receiving a data element in the format of the first data model of the first software component, translating the data element from the format of the first data model of the first software component to the format of the third data model in the data wedge and storing the translated data element in the data storage, by the data wedge (e.g. Note: UML structure storing 2 XML-based data representation of business model in terms of business class or model blocks – see para 0239 ,pg. 13 - reads on third model and storage of thereof – Fig. 58-68 – as a result of mappings from the XMulator); and

retrieving the data element from the data storage and translating the data element into the format of the second data model of the second software component by the data wedge (e.g. *language 1, language 2* – para 0034-0035, pg. 3; *a new XML comes along … adapted for the* *new language* - para 0070, pg. 5; para 0039, pg. 3; para 0237-0240) after receiving a request for the data element from the second software component.

As per claim 19, Worden discloses reading the data element translated into the format of the second data model by the second software component (e.g. *converts from class model back to language 2* – para 0035, pg. 3; para 0039, pg. 3).

As per claims 21-22, Worden discloses creating an instance of the data wedge (e.g. Fig. 12 – Note: each session per application or agent with login reads on instance) ; wherein the first and second schemas further comprise a name of the data wedge (e.g. XMulator Ltd 2000, Fig. 12).

As per claim 23, Worden discloses wherein integrating the first schema into the data wedge includes setting default data elements and data values for the first data model (e.g. *empty map tree ... basic tree* – para 0473, pg. 25 – Note: basic tree reads on defauft value for first data model; para 0085-0094, pg. 6; Fig. 17, 20-23 – Note: purchase model to be implemented with W3C template form of a extensible language **reads on** W3C/XML basic format having default and values to be extended with associations or relations from a UML, during the session wherein first software component data are integrated into the XMulator) of the first software component.

As per claim 24, Worden discloses retrieving the data element from the data storage (*database* - para 0436-0439, pg. 24) and translating the data element from the format of the third data model to the format of the first data model of the first software component by the data wedge (e.g. para 0028-0029, pg. 2; para 0037, pg. 3) after receiving a request for the data element from the first software component (e.g. bridge between meaning and structure … for each XML language he uses, language designer – para 0061, 0063pg. 5 – Note: language designer submitting his XML-based design in order to obtain model structure and mappings **reads on** data wedge transforming XML into model mappings upon developer request – see *takes as an input a document ... such as DOM ... answering the query* – para 0044, pg. 4).

As per claim 25, Worden discloses a computer system for translating data between a format of a first data model of a first software component and a format of a second data model of a second software component (e.g. para 0034, 0035, 0039, pg. 3), the system comprising: a processor; and a memory coupled to said processor, the memory having stored therein data and sequences of instructions which, when executed by said processor, cause said processor to:

create a first schema comprising the first data model of the first software component; create a second schema comprising the second data model of the second software component (refer to claim 1);

create a third data model and a data storage in a data wedge by integrating the first schema and the second schema into the data wedge (refer to claim 1, in view of the USC 112 1st paragraph);

receive a data element in the format of the first data model of the first software component, translate the data element from the format of the first data model of the first software component to a format of the third data model in the data wedge and store the translated data element in the data storage (refer to claim 1); and

retrieve the data element from the data storage and translate the data element into the format of the second data model of the second software component after receiving a request for the data element from the second software component (refer to claim 1).

As per claim 28, refer to claim 21.

As per claim 29, refer to claim 24

As per claim 30, refer to claim 22.

As per claim 31, refer to claim 23.

As per claim 32, Worden discloses a computer system for translating data between a format of a data model of a first software component and a format of a data model of a second software component (refer to claim 25), the system comprising:

a processor; and a memory coupled to said processor, wherein said processor is configured to execute a sequence of instructions contained in said memory, the instructions comprising

a data wedge (e.g. XMulator - Fig. 12, 14-71) including a first schema of the first software component and a second schema of the second software component (XML … first language … second language – para 0033-0035, 0039, pg. 3),

the data wedge configured to translate a data element from the format of the data model of the first software component in accordance with the first schema to a data model of the data wedge (para 0038, 0040 pg. 3; para 0063, pg. 5; *first XML based language .. XML logical structures ... information model* – para 0039, pg. 3 Note: UML structures – Fig. 2, Fig. 58-68 - storing user provided XML-based data representation – first and second data model/schemas- of business model in terms of business class or model blocks – see para 0239, pg. 13 - **reads on** data model in accordance to each such schemas and storage of thereof – Fig. 58-68 – as a result of mappings from the XMulator) and

when a request is received from the second software component, translate the data element from the format of the data model of the data wedge to the format of the data model of the second software component in accordance with the second schema (e.g. *language 1, language 2* – para 0034-0035, pg. 3; *a new XML comes along ... adapted for the new language* - para 0070, pg. 5; para 0039, pg. 3).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) a patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 18, 20, 26-27, 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Worden, USPuBN: 2003/0149934.

As per claim 18, Worden does not explicitly disclose triggering an event to notify the second software component of the availability of the data element received from the first software component and stored in the data wedge. Worden discloses GUI interface to return mapping results to user, making it visible to all users (see Fig. 31, 74, 75, 78-80 – Note: Gui depicting mapping returns **read on** event visible to users; *data source ... visible to all users* – Fig. 74 ;Fig. 32-35; Fig. 38-52). Based on Gui events and notification of results made available to all users, the concept that another user (pertinent to second software component) being informed from availability of mappings results from processing XML source based on database queries is disclosed. It would have been obvious for one skill in the art at the time the invention was made to implement the Gui interface by Worden, so that notification of available mappings from a first software component (or first user) would be implemented by as trigger in the GUI by

Worden, thereby enabling immediate information availability to other user (second software component) as suggested above, for enabling users pertinent to the second software component to make use of the mappings and incorporate its into their application in another language (see para 0035, pg. 3; *sets of mappings ... document in the first XML based language ... to be translated ... to a document in the second XML based language ... for the two languages* – para 0039, pg. 3).

As per claim 20, Worden does not explicitly disclose: removing an obsolete data element from the first data model of the first software component and causing the data wedge to remove the translated data element from the third data model. However Worden discloses agent and user paradigm wherein the XMulator provides MDL matching, XML-to-XML mappings as displayed results; and transformation of schemas from a first user into XSLT to return to the user of the second user/software component also using the XMulator tool (see para 0260 to para 0264 – pg 14-15; Fig. 9; Fig. 54-56; para 0842-0858, pg. 37-38) including removing of mappings data (see *delete* button – Fig. 29-31; para 0564, pg. 29; para 0720, pg. 34). It would have been obvious for one skill in the art at the time the invention was made to implement the agent and the XMulator interface in Worden so that in the (third model) mappings interface, the Gui-implemented delete/remove function would be for removing a XML component mappings as in removing of an obselete data element in the XML of the first software component using the wedge created third model as set forth in the mappings displayed above, to enhance correctness of the model with more updated mappings.

As per claims 26-27, refer to the rationale of claims 18, 20.

As per claim 33, refer to the rationale as set forth in claim 18.

/****************/

(10) Response to Argument

USC § 112, 2nd paragraph Rejection.

(A) Appellant has submitted that paragraph 22 of the Specifications teaches Data Wedge to minimize code modification by 'utilizing a data model of cooperating components' when 'integrating software components', such that the 'data model' is the third model that 'integrates' multiple 'software components' (Appl. Rmrks pg. 7, 2^{nd,} ^{3rd} para); and further has explained that each client creating of a schema constitutes a "software component" defining a 'logical data model' to be submitted to the Wedge upon connection (Specifications paras 37-38); that is, each 'software component' submitted logical model (definition as being a superset of XML – para 24 of Specifications) is such that it is integrated dynamically into that Wedge: Appellant has deducted that the Appellant's disclosure has indeed reasonably conveyed teaching that schema of data model from each connected client is integrated into the Wedge.

First, utilizing a Wedge to integrate schemas from different connected clients is not the language being deemed lacking disclosure support. Claim 17 recites 'creating a third data model' is achieved *by integrating* first schema and second schema *into the Wedge*; the claim language and the Specifications have not stated that the Wedge and the created "third data model" is interchangeable entity. Indeed, the Wedge is understood as a shared network console tool serving as integrating environment to enable concurrently connected users to submit the representation of their business logic (including data and relationship respective to each client defined schema – Schema A, Schema B, Figure 1) and as a result of the parsing and transformation by the Wedge -- a server component that compiles schema and exposes interfaces for client usage (see para 22, pg. 6), each client can have a view of his defined data and logic in terms of tree, such that each tree view is performed by the integrating aspect of the Wedge (and its physical storage functionality) such that each view is respective to each user, and different from one user to another (see Data View A, Data View B). Clearly, the integrating functionality of the Wedge amounts to a server environment having GUI interface, storage, parsing capability and exposing of tree views, not a "third data model" that is created by the Wedge and contains integrated data coming from different Schema A, or Schema B submitted by each connected client. One of ordinary skill in the art cannot see where in paragraphs 022, 023, 037, 038 that because the Wedge dynamically integrates schema's data (from separate clients) it is evidenced that the Wedge becomes a "third data model" being created data integration as exactly claimed. In other words, the claim requires that by integrating first and second 'schema' (from first and second software component) into the Wedge, a <u>third</u> DATA model is formed. The above paragraphs so-proffered by Appellants amount to describing the integrating functionality of a "Wedge" serving a shared console to let each user have an interfacing avenue in order for each to visualize his/her own user-defined logic (e.g. via client-specific view of **data** tree; see *Wedge 10 ... provides a different view .. .to every client or component* – para 0023, pg. 6) whereby to modify its data or functionality. It is largely inconclusive as to whether these paragraphs clearly show creating of a 'third data model' based on integrating first and second schema; nor is there clear expressing that after receiving first and second schema, the Wedge creates a modified version of itself to instantiate a 'data version' of the WEDGE in terms of "third data model' having included therein data from the submitted schemas, because in light of paragraphs 022, 023, 037, 038 the Wedge remains a integrating tool and does not transform received data into a 'data model' as a result of capturing first and second client submitted schema logic.

Second, the 'Summary of the Claimed subject matter' ((5) of the Brief) compounds the inconclusive support by the Specifications by citing paragraph 38, pg. 10. According to this paragraph, any schema detected as having been previously instantiated by the Wedge with a same name, will be automatically integrated with that previous instance. Clearly, one schema from one specific 'software component' with same name instantiation (para 038) cannot be consistent with paragraphs 022, 023 where multiple users concurrently submitting their own schema can have the Wedge expose a different view of their respective schema data via the Wedge share console or COM functionality; the support for 'integrating' is understood in two different contexts, none of which clearly enabling one of ordinary skill in the art to learn how the Wedge tool, when having integrated therein client A, and client B schema data (Fig. 1), actually creates a "third data model" combining the different schema, especially when the Wedge as disclosed (para 0021-0022) is merely a server component (or wedging tool) that captures external NW data sources into internal data thereby expose interfaces for users to use their respective data or to share data. None of the paragraphs proffered by Applicants spells out that the Wedge is actually the 'third data model' -- newly formed/instantiated after first schema and second schema have been translated and integrated by the tool-- in light of separate schema being turned into different view pertinent to respective client logic. Nor is there evidence that each logical tree (DATA VIEW A, B – Fig. 1) constitutes a *third data model* because that would defeat the conviction about formation of integrated data as argued by the Appellant.

A *integrating tool* is construed as integrating inputs from disparate components to provide a exposed data presentation based on the definition parsed therefrom; and in exposing data, graphical views or data tree can be generated by the tool to support the development endeavor of the users using the very tool; that is, the tool (e.g. COM framework) can generate a view comprising of data extracted from submitted software components, such that the so-generated view **is not the tool** but a product from the tool. The different views based on the "wedging" functionality of the Wedge (see para 0022, 0023) amount to one such visible data view and clearly (see Figure 1 of the Disclosure) each such view does not include data from other schema (emphasis added) because each user can modify the data at it is returned by the Wedge (see para 0040-0041) for implementing user-specific development. Nor can adding one instance of user-specific schema to an existing schema of identical name (para 038) constitutes integrating two users schemas to yield a *third data schema* as claimed, each schema coming from respective 'software component', not coming from one same client. Nor the capacity to retain schema and previously instantiated structure by the Wedge environment (see para 0043) constitutes a authentic 'data model' dynamically created for modeling separately submitted schema logic/definitions from multiple users.

Based on the lack of OR inconsistency in textual support as to how based on 'integrating first and second schema' a "third data model" is formed within the Wedge, the creating of 'third data model' is deemed not provided by the Disclosure in terms that would facilitate one of ordinary skill in the art to make or use the invention. In view of the interpretation of the Disclosure and the concept of 'integrating tool' set forth above, one would have to resort to undue experimentation to implement "third data model' in a Wedge by incorporating first schema and second schema from concurrent users, in terms of generating a merge or integrated 'third data model', because of the lack of description in accordance with the disclosed portions proffered in the "Summary of Claimed Subject matter' or the entire disclosure.

The argument is deemed not sufficient to overcome the rejection.

(B) Appellant has submitted that as each client provides schema, that client component "is not a person" but actually disclosed as 'software component' (Appl. Rmrks pg. 8-9) rendering 'creating' work done by the Invention. The argument denies that the schema comes from a person but rather enforce that a 'client' involved in the "creating" is a software component. When the 'Wedge' (purported to reduce source code modification by developing users – para 002; point of sale – para 006, 010) is serving a community of users within a network encompassing connected host computers with input device (see Figure 3; *personal computers* - para 0044-0046), given the portability nature of XML format enabling NW computers to interact with a server 300 (e.g. COM tool) whose functionality is to provide views as disclosed, it is hard to construed that client users are reduced to mere "software components", as argued, in the sense that the one same "software component", respective to a host computer accessing the server (see para 003), creates (CREATE - Figure 2) or 'must create' a schema (para 024, pg. 6) and then be presented a visible DATA VIEW A/B (Figure 1) exposed/generated by the COM/Wedging console. There is no part of the Disclosure that clarifies that the client for just *creating a schema* is hands-free auto-run software whereas the client component being exposed with a tree view (Fig. 1) is actually a person or human eye. The statutory aspect of an invention is to provide a clear *utility* serving human kind, and interconnected client host computers to make use of views provided from a COM service leads to one to-be-expected endeavor of serving human-operated computers for human developing of code; and accordingly, creating schema done by a human user cannot be part of the invention. The argument states however, that a client is 'software component'. **Admitting that the client** (e.g. a subscriber – see para 0041, pg. 11) who is seeking support from a Wedge server (Drawings: populate model - Fig. 2) to be presented DATA views for further modification (i.e. whereby enabling a simplified development approach as to reduce source code implementation or to achieve a *desired functionality* – see Background of invention, pg. 1; para 0053, pg. 14) **is NOT a person/human being** entails that subscriber components at host computers (Drawings: Figure 3) communicating via a session with the Wedge server (for applications like point of sale, transaction - para 006) are automated client software components that would be running without human intervention and directly responsible for creation of XML definition then decide/assess on what to modify when exposed with the tree view (Fig. 1, 2). In this regard, there is no sufficient description in the Disclosure to attest to the above 'software components' acts of connecting to a server, creating XML, viewing and populating a tree (emphasis added), and deriving what to modify from that tree view (presented by COM Wedge) being executed in strictly software; that is, no corroboration from the Disclosure that subscriber components have been automated in software-alone implementation, which would operate on its own (emphasis added) from Server connection (including exchanging NW messages) to modifying the view (e.g. Fig. 2) thence achieving a *desired functionality* within a business (point-of-sale) application. Software alone without support of hardware would also fall into a non-statutory subject matter, while software automation without proper description as to how it has been implemented (as to obviate any human role) would raise a USC 112 1st paragraph deficiency.

Hence, the ‘creating' of schema is not enabled by the Disclosure in 2 manners. On one hand, if the client amounts to automated software, a lack of description (USC 112 first paragraph) according to that interpretation is deemed blatant. That is, no where is there description of automated component/module that runs to connect, create schema, and decides on how much to implement code modification or further request/access point-of-sale data.

On the other hand, if creation of XML is provided by a network human user within a point-of-sale context, the mere fact of defining a XML by a transaction user using a GUI console at the host computer prior to sending it to the session-connected Wedge indicates that the ‘creating’ is not work provided by the Inventor; i.e. deficiency for a non-enabled claim language.

The argument that ‘creating’ is work done by the Inventor via the ‘client' implemented by the Invention as ‘software components’ is deemed far-fetched and unjustified in more than one respects. The argument is deemed insufficient; the rejection will not be overcome and the ‘creating’ limitation will not be given full patentable weight.

USC § 102(b) Rejection

(C) Appellant has submitted that ‘syntactic and information for intended languages” in Worden cannot be viewed respectively, as schema being “data models” for respective “software components” (Appl. Rmrks pg. 10 top). Applications written via a representing language can be construed as “software components" on host machines, ‘software components’ viewed as source of the individual user XML being submitted to Worden mapping framework (or Wedge), applications using a B2B/XML standards of the like of business applications/transactions whose implementation can be one target language (or through the interoperability aspect of XMulator's mapping, can be derived from another language via portability of XML – see Worden: para

0006-0016, pg. 1-2). The claim does not provide details surrounding this 'software component' in order to obviate B2B applications (one user application that sends XML schemas) from fulfilling this language as recited. Further, the first schema and second schema are analogized to the XML specifications from users involved in business transactions – see para 0090-0097 – and the transaction-related applications for which the XML or MDL structured language is created reads on first/second data model or schema whose source are 'software components' since the pertinent business user's XML schemas as submitted to the schema-mapping framework or XMULator are used for or support model mappings against equivalent class components – Fig. 2, Fig. 58-68 – in order to implement user's application.

Hence, Worden has disclosed first and second schema submitted to a XMuLator framework that provides model mappings using the very XML as inputs (or modification thereof – MDL), each such XML is representative of a business application or programming language desired/targeted by the user (XSLT or Java code – see para 0323-0324; 0343-0344) the XML mapping yielding model class/objects (or third data model; i.e. data hierarchy visible to the respective user – see USC 112, Rejection) as viewed in Worden's graphical constructs (Figure 58-68 ; *business information model* in Java classes - para 0324-0330, pg. 18). It is therefore reasonable to say that the *first and second schemas* for respective 'software component' is disclosed in the Worden's framework as cited; and the argument is non-persuasive to overcome the rejection.

(D) Appellants have submitted that Worden as cited does not disclose 'storing the translated element … by the wedge' (Appl. Rmrks pg. 10, 2nd para). The mapping based on user submitted schema (or message received) for enabling the XMULator in performing mappings whereby retranslating of original language (via the portability of XML format defining model objects) to a desired target language (Java code, XSLT, or another XML file) entails storage of data received and model constructs derived from the mappings within the very framework. The claim does not depict storage in a more elaborated manner (e.g. using database record or library of legacy metadata) to preclude Worden cited portions from being used. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

The allegation that the XML-based translation in Worden would not enable a person of ordinary skill in the art to make Applicants' claimed subject matter (Appl. Rmrks pg. 10 bottom) is construed as a mere allegation without prima facie pointing out of specific distinguishing traits between Worden's cited parts in relation to a specific claim language.

USC § 103(a) Rejection

(E) Appelant has submitted that based on the analysis of the 102 Rejection, Worden fails to show or suggest all the element of the independent claims, hence fails to teach or suggest the claimed elements depending upon those independent claims (Appl. Rmrks pg. 11). This global allegation is referred back to the sections C, D set forth from above, where it is shown that none of the arguments against the rejection for the independent claims stands sufficient to overcome the rejections.

/************************/

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Tuan Anh Vu /

Primary Examiner

02-23-2010

Conferees:

/Lewis A. Bullock, Jr./

Supervisory Patent Examiner, Art Unit 2193

/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191